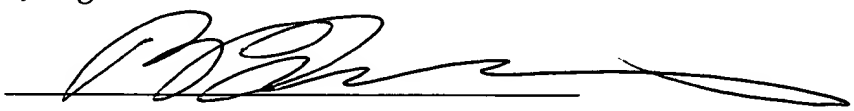
REMARKS

The purpose of the Preliminary Amendment is to correct the typographic errors and to clarify and emphasize the advantages of this invention. No "New Matters" are added to the Application as originally filed on July 31, 2001.

The applicant hereby respectfully requests that Examiner would enter the preliminary Amendment to correct the Specification as that originally filed on July 31, 2001.

Respectfully submitted,
Dongyi Jiang, et al.

By 

Bo-In Lin -- Attorney, Registration No. 33,948
13445 Mandoli Drive, Los Altos Hills, CA 94022
(650) 949-0418 (Tel), (650) 949-4118 (Fax)

Version with Markings to show Changes Made

In the Specification:

I. Please amend the Specification as set forth below:

a) On page 1, lines 12 to 16, please amend the Field of the Invention as set forth below:

The present invention relates to computer network security. More particularly, this invention is related to fast table-lookup algorithms of multiple-dimensional sequential data array for broad ranges of applications. The applications may include firewall, i.e., a combination of computer hardware and software for selectively accepting network data communications and rejecting unacceptable data transmissions to safeguard a computer network based on a predefined policy table.

b) On page 3, in lines16 to 25, please amend the third paragraph as set forth below:

Packet filters are typically configured in a "default permit or denial stance", i.e., that which is not expressly prohibited/permitted is permitted /prohibited. In order for a packet filter to prohibit potentially harmful traffic, it must know what the constituent packets of that traffic look like. However, it is virtually impossible to catalogue all the various types of potentially harmful packets and to distinguish them from benign packet traffic. The filtering function required to do so is too complex. Hence, while most packet filters may be effective in dealing with the most common types of network security threats, this methodology presents many chinks that an experienced hacker may exploit. The level of security afforded by packet filtering, therefore, leaves much to be desired.

b) On page 5, in lines 1 to 7, please amend the first paragraph as set forth below:

In general, network firewalls employ filter rules or policies to police network communication. In such implementation, a data packet is examined and

checked with fire filter policy rules. In essence, the policy lookup in the network firewall is to find an efficient way to map a [four] five-dimensional space DA, SA, DP, SP and protocol, to one dimension policy space. Historically, most firewalls use linear search algorithms. These algorithms are very time consuming and [without] with O(N) as the upper bound of searching time and the searching time increase linearly as the Policy List growing.

d) On page 10, in lines 1 to 17, please amend the first paragraph as set forth below:

Referring to Fig. 1 again, each entry of the policy table is assigned a policy entry counter ip = [0,] 1, 2, 3, ...N, according to an ascending sequential order starting from zero (step 135) where N is the total number of policy entries in the policy table. The process continues by assign an policy entry counter ip to each table entry corresponding to every {SASN, DASN} pair in the source-destination address mapping table and each table entry corresponding every {SPSN, DPSN} pair in the source-destination port mapping table (SDPMT) (step 140). All the table entries are initially registered as "unused" before the policy entry counter ip is entered in either the SDAMT or the SDPMT tables, and each table entry in either of these two tables is entered only with the first ip counter. Once a policy entry counter ip is entered for a table entry, that table entry in either the SDAMT or SDPMT tables is assigned with one unique ip counter and will not be changed unless overwritten by other procedure when there are changes made to the policy table. A mapping process is then carried out to transform from the four dimensional space defined by four entries of ip in four tables, i.e., SDAMT and SDPMT , to another two dimensional space represented by a policy mapping table (PMT) (step 145).

e) On page 10, in lines 18 to page 11 line 1, please amend the second paragraph as set forth below:

Referring to Figs. 5A to 5C for an example for illustrating the mapping process to construct the policy-mapping table. Figs. 5A and 5B shows the SDAMT and SDPMT entries at the time when the processes for constructing these two tables are completed for the policy entry counter ip=4. For policy-entry counter ip=1, examining Figs. 5A and 5B, there is only one combination, i.e., {1, 1}. An ip counter number, i.e., ip = 1, is entered into the slot {1, 1} of the

policy mapping table (PMT). For ip = 2, there are possible combinations of {1, 2} and {2, 2}. An ip counter number, i.e., ip = 2, is entered into the slot [{3, 1}], {1, 2}, and {2, 2} of the policy mapping table (PMT). For ip = 3 there are possible combinations of {3, 1} and {3, 3}. An ip counter number, i.e., ip = 3, is entered into the slot {3, 1}, and {3, 3} of the policy mapping table (PMT). For ip = 4, the possible combinations are {4, 2} and {4, 4}. An ip counter number, i.e., ip = 4, is entered into the slot {4, 2}, and {4, 4} of the policy mapping table (PMT). The X-Y coordinates on the PMT table are therefore generated by combining the policy entry counters from the source-destination address mapping table (SDAMT) as the X-coordinate, and the policy entry counters from the source-destination port mapping table (SDPMT) as the Y-coordinate for all policy entry counter ip = 1, 2, 3, ..., N, a policy mapping table is formed. A two two-dimensional tables are mapped into a two dimensional policy mapping table as that illustrated in Fig. 5C.

f) On page 11, in lines 3 to 122, please amend the second paragraph as set forth below:

Referring back to Fig. 1 again, for the purpose of effectively conducting a "fast policy lookup" process, four "balanced binary trees" are structured (step 150). These four binary trees are a source address tree, a destination address tree, a source-port tree and destination-port tree. These balanced binary trees provide the benefits that the table-lookup processes can be more expeditiously completed because the processes are performed in a more structured, organized and balanced manner. The search time is reduced from O(N) for the unstructured array to O(lnN) when balanced binary trees are implemented. Suppose that there are N source and destination addresses and M source and destination port, the process generally starts from a root [of] represented by a source/destination address sequence number of N/2 and source/destination port number of M/2. Each binary tree starts with a root N/2 or M/2, each having two branches having the source-destination address and port sequence numbers [of] starting from [(N/2-1), (N/2+1)] and [(M/2-1), (M/2+1)] respectively. In receiving an incoming packet, the header of the packet is parsed to get the source/destination addresses and source/destination port number (step 155). These address and port number are then applied to travel down the four binary trees to find the source/destination address sequence numbers, i.e., SASN and DASN, and the

source-destination port sequence number, i.e., SPSN and DPSN (step 160). Using the SASN and DASN as X-Y coordinates, a policy entry counter ip(A) is determined from the SDAMT as that shown in Fig. 5A. Using the SPSN and DPSN as X-Y coordinates, a policy entry counter ip(P) is determined from the SDPMT as that shown in Fig. 5B (step 165). These two policy entry counter numbers ip(A) and ip(P) are then used as X-Y coordinates to lookup the final policy entry counter number from the policy mapping table as that shown in Fig. 5C (step 170).

g) On page 18, in lines 1 to 2, please amend the first paragraph as set forth below:

The computational complexity of policy lookup is reduced from [0] $\underline{O}$(n) to O(lgn), where the n is the [length] number of entries of the Policy List.